ÖAW Austrian Academy of Sciences

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

JKU JOHANNES KEPLER UNIVERSITY LINZ

Fat arcs for implicitly defined curves

Szilvia Béla Bert Jüttler

DK-Report No. 2009-08 11 2009

A–4040 LINZ, ALTENBERGERSTRASSE 69, AUSTRIA

Supported by

Austrian Science Fund (FWF)

FWF Der Wissenschaftsfonds.

Upper Austria

LAND OBERÖSTERREICH

DK sponsors:

- **Johannes Kepler University Linz** (JKU)
- **Austrian Science Fund** (FWF)
- **Upper Austria**

Fat arcs for implicitly defined curves*

Szilvia Béla[1] and Bert Jüttler[2]

[1]Doctoral Program in Computational Mathematics and
[2]Institute of Applied Geometry,
Johannes Kepler University, Altenberger Str. 69, 4040 Linz, Austria

Abstract. We present an algorithm generating a collection of fat arcs which bound the zero set of a given bivariate polynomial in Bernstein–Bézier representation. We demonstrate the performance of the algorithm (in particular the convergence rate) and we apply the results to the computation of intersection curves between implicitly defined algebraic surfaces and rational parametric surfaces.

1 Introduction

Bounding geometric primitives which enclose segments of planar curves are frequently needed for various geometric computations, e.g., for solving the intersection problem between two planar curves. Axis-aligned bounding boxes (min-max boxes), which can easily be generated both for planar parametric curves and for implicitly defined curves, are one of the simplest instances. Other useful primitives include fat lines (bounding strips, see e.g. [1]), the convex hull of the control polygon, or fat arcs [2].

The performance of a bounding primitive depends on the approximation order. For a bounding primitive with approximation order k, the number of primitives needed to bound a curve with a given tolerance ε grows like $\sqrt[k]{1/\varepsilon}$. Consequently, the use of geometric primitives with higher approximation order may provide computational advantages. Bounding boxes have only approximation order $k = 1$, while both the convex hull of control polygons and fat lines provide approximation order 2, and fat arcs even have approximation order 3.

Clearly, it is possible to define bounding primitives with an even higher approximation order. Fat arcs seems to be particularly useful since they provide a reasonable trade-off between geometry flexibility and the computational simplicity of elementary geometric operations. For instance, the computation of the intersection of two circular arcs requires solely the solution of quadratic equations, while this becomes far more complicated for higher order objects.

Various methods for generating an arc spline curve which approximate a given parametric curve with a prescribed tolerance have been described in the literature, see e.g. [3] for many related references. The use of arc splines for geometric design applications can be traced back to a classical VTO report of

* This work was supported by the Austrian Science Found (FWF) through the Doctoral Program in Computational Mathematics, subproject 3

Sabin [4]. Marciniak and Putz dealt with the minimization of the number of arcs to approximate a curve under a give tolerance [5]. Later Qiu et al. improved their method [6]. In a number of papers, Meek and Walton applied arc splines to approximate parametric curves [7–9]

Yong used arc splines for quadratic Bézier curve approximation [10]. Feichtinger et al. compared various biarc interpolation schemes [11]. Held and Eibl approximated with biarcs simple planar polygons either for symmetric and asymmetric tolerance bounds [12].

Fat arcs as bounding geometric primitives were introduced by Sederberg [2]. Algorithms which generate bounding fat arcs for parametric curves are described, e.g., in [13]. Of course, any arc spline approximation technique can also be used to generate bounding fat arcs, simply by offsetting the obtained curve. The existing techniques for fat arc generation deal exclusively with the case of parametric curves.

In this paper we present an algorithm which generates a collection of fat arcs bounding an implicitly defined curve with a prescribed tolerance. First we describe how to find a fat arc for a single curve segment in a box. Then we combine this technique with adaptive subdivision in order to find a global approximation. As an application, we apply the fat arcs to approximate the intersection curve between implicitly defined and parametric surfaces.

2 Preliminaries

We recall the construction of fat arcs for parametric curves. In addition, we present a result concerning distance bounds and a criterion which guarantees single arcs.

2.1 Approximation by arcs and fat arcs

In the case of planar parametric curves, the construction of fat arcs has been discussed in [13, 2]. The methods described there generate an approximating arc which possesses a finite thickness. First, a median arc through three points of the original parametric curve segment is defined, see Fig. 1. For instance, these three points can be chosen as the two endpoints of the curve segment and the intersection point of the curve and the bisector of the endpoints. As the second step, the method computes (an upper bound of) the distance between the original curve segment and the median arc. Finally, an offset of the median with this distance is defined. The boundaries of the offset are concentric arcs, whose radii are the sum and the difference of the median arc radius and the distance of the curves. They define a bounding domain for the original curve segment. This bounding region is called a *fat arc*. Since the approximation order of circular arcs is equal to three, the offset distance behaves as $O(h^3)$, where h is the length of the given curve segment (or of its parameter interval).

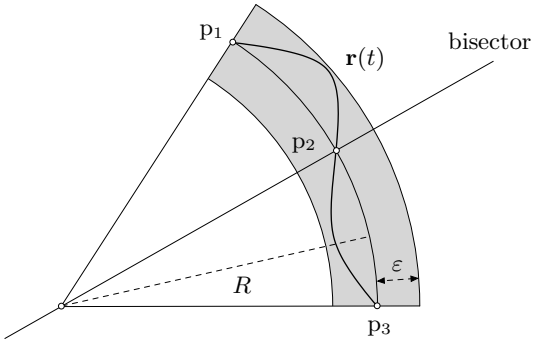

Fig. 1. Fat arc generation

2.2 Distance estimate

In order to construct fat arcs for an implicitly defined curve, we shall use the properties of the defining function. We assume that the bivariate polynomial is given by its tensor-product Bernstein-Bézier (BB) representation of degree (m, n):

$$f(\mathbf{x}) = \sum_{i=0}^{m} \sum_{j=0}^{n} d_{ij} \beta_i^m(x_1) \beta_j^n(x_2), \tag{1}$$

with respect to the domain $\Omega = [0, 1]^2$ and with certain coefficients $d_{ij} \in \mathbb{R}$, where

$$\beta_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad t \in [0, 1]. \tag{2}$$

More generally, (x_1, x_2) will also be used as local coordinates

$$x_1 = \frac{\xi_1 - a_1}{h}, \quad x_2 = \frac{\xi_2 - a_2}{h}, \tag{3}$$

with respect to the general axis-aligned box $B = [a_1, a_1 + h] \times [a_2, a_2 + h]$ of size $h \times h$ in the (ξ_1, ξ_2)-plane.

The implicitly defined curve is given as the zero set of the bivariate polynomial,

$$\mathcal{F} = \{\mathbf{x} \,:\, f(\mathbf{x}) = 0 \ \wedge \ \mathbf{x} \in [0, 1]^2\}. \tag{4}$$

Clearly, the curve may be empty, or it may consist of more than one curve segment.

In order to construct a fat arc, we need to bound the distance between the median arc and the curve using a result from [14], see also [15]. On the one hand, we consider the medial arc as a parametric curve $\mathbf{g} : t \mapsto \mathbf{g}(\mathbf{t})$ with parameter domain $t \in [a, b]$, which traces the point set

$$\mathcal{G} = \{\mathbf{g}(t) \,:\, t \in [0, 1]\} \tag{5}$$

where we assume that $\mathcal{G} \subset [0,1]^2$. On the other hand, in order to avoid certain technical difficulties, we consider the set

$$\mathcal{F}^* = \mathcal{F} \cup \partial\Omega \tag{6}$$

which is obtained by adding the boundary of the domain to the curve $\mathcal{F}$. The one-sided Hausdorff distance of $\mathcal{F}^*$ and $\mathcal{G}$ is defined as

$$\mathrm{HD}(\mathcal{G}, \mathcal{F}^*) = \sup_{t\in[0,1]} \inf_{\mathbf{x}\in\mathcal{F}^*} \|\mathbf{x} - \mathbf{g}(t)\|. \tag{7}$$

We recall the following result from [14]:

Lemma 1. *If there exist positive constants c, η such that*

$$\forall \mathbf{x} \in \Omega: \quad c \le \|(\nabla f)(\mathbf{x})\| \quad \textit{and} \quad \forall t \in [0,1]: \quad |(f \circ \mathbf{g})(t)| \le \eta \tag{8}$$

hold, then the one-sided Hausdorff distance is bounded by

$$\mathrm{HD}(\mathcal{G}, \mathcal{F}^*) \le \frac{\eta}{c}. \tag{9}$$

Consequently, the parametric curve is contained in ε–neighborhood of $\mathcal{F}^*$, where $\varepsilon = \eta/c$. However, it should be noted that this distance bound does not guarantee that the implicitly defined curve is then also contained in an ε–neighborhood of the parametric curve. The algorithm presented below uses an additional test to guarantee this property. Nevertheless, in all computed examples the above distance bound provided a safe and conservative estimate for the two-sided Hausdorff distance of the implicitly defined and the parametric curve.

2.3 Evaluation of the constants

In order to find the constants c and η in Lemma 1, we represent the median arc as a quadratic rational Bézier curve,

$$\mathbf{c}(t) = \sum_{i=0}^{2} \mathbf{c}_i \frac{\tilde{w}_i \beta_i^2(t)}{\sum_{j=0}^{2} \tilde{w}_j \beta_j^2(t)}, \quad t \in [0,1]. \tag{10}$$

The composition $f \circ \mathbf{c}$ is a rational function of degree $2(m+n)$ which can be represented by its BB representation with certain coefficients d_i and weights w_i. The weights are found by evaluating the $(m+n)$th power of the denominator in (10). If all weights and coefficients are positive, then

$$|(f \circ \mathbf{c})(t)| = \sum_{i=0}^{2m+2n} \frac{d_i w_i \beta_i^{2n+2m}(t)}{\sum_{j=0}^{2m+2n} w_j \beta_j^{2m+2n}(t)} \le \frac{\max\limits_i d_i w_i}{\min\limits_i w_i} = \eta. \tag{11}$$

In order to find the second constant c, we generate the tensor–product BB representation

$$\left(\frac{\partial f(x_1,x_2)}{\partial x_1}\right)^2 + \left(\frac{\partial f(x_1,x_2)}{\partial x_2}\right)^2 = \sum_{i=0}^{2m}\sum_{j=0}^{2n} h_{ij} \beta_i^{2m}(x_1) \beta_j^{2n}(x_2) \tag{12}$$

which can be found using differentiation, product, and degree elevation formulas, see [16]. If all coefficients h_{ij} are positive, then

$$\|\nabla f(\mathbf{x})\| \geq \sqrt{\min_{i,j} h_{ij}} = c. \tag{13}$$

2.4 Identifying boxes with single segments

In order to generate a fat arc approximation in a box, we need to ensure that the box contains only one segment of the implicitly defined curve segment. Various criteria for isolating a regular segment of an algebraic curve have been discussed in the literature. For instance, different types of discriminating curve families have been used in [17]. These discriminating families are particularly useful in combination with algorithms that trace the algebraic curve segments.

In our case, we are interested in a criterion which guarantees that the box Ω contains a single curve segment with exactly two intersections with the boundaries.

Lemma 2. *Consider an algebraic curve segment defined by the polynomial* (1)*. We say that the coefficients exhibit a* ***corner event****, if*

- *the coefficient at one of the corners is equal to zero and*
- *the first non-zero coefficients along the two neighboring boundaries have a different sign.*

We say that the the coefficients exhibit an ***edge event****, if*

- *the control polygon along one the box boundaries has exactly one sign change from plus to minus or vice versa.*

If the number of the corner and edge events is equal to two and if the positive constant c of Lemma 1 exists, then the box contains a single curve segment, which is regular, connected, and which intersects the box boundary in exactly two points.

For the proof it suffices to observe that each event guarantees that the implicitly defined curve crosses the box boundary in exactly one point. Moreover, the curve does not contain any closed loops, since the gradient vector does not vanish in Ω.

The conditions of Lemma 2 are sufficient, but not necessary. For example, the lemma excludes the case of a single arc of the implicitly defined curve crossing the same edge twice.

3 Approximation of a single curve segment

This section focuses on the approximation of a single segment of the algebraic curve. We describe the algorithm and demonstrate its performance.

Algorithm 1 `FatArcSegment`(f, B, ε)

Require: The conditions of Lemma 2 are satisfied.

1: $b = \{\mathbf{b}_1, \mathbf{b}_2\} \leftarrow$ approximate boundary points of the implicitly defined curve
2: $c = \{\mathbf{c}\} \leftarrow$ approximate inner point of the implicitly defined curve
3: **if** $\#b = 2$ and $\#c = 1$ **then**
4: $\quad \mathcal{M} \leftarrow$ circle through $\mathbf{b}_1, \mathbf{c}, \mathbf{b}_2$ {median circle}
5: $\quad d \leftarrow$ upper bound of $\mathrm{HD}(\mathcal{M} \cap B, \mathcal{F}^*)$ {see Lemma 1}
6: $\quad$ **if** $d \leqslant \varepsilon$ and $d \leqslant$ radius of $\mathcal{M}$ **then**
7: $\quad\quad \mathcal{C}_d \leftarrow$ offset ring of $\mathcal{M}$ with distance d {fat circle}
8: $\quad\quad \mathcal{C}^+, \mathcal{C}^- \leftarrow$ inner and outer circle of $\partial\mathcal{C}_d$
9: $\quad\quad$ **if** there is no sign change of f along $\mathcal{C}^+ \cap B$ or $\mathcal{C}^- \cap B$ **then**
10: $\quad\quad\quad$ **return** $B \cap \mathcal{C}_d$ {fat arc has been found}
11: $\quad\quad$ **end if**
12: $\quad$ **end if**
13: **end if**
14: **return** $\emptyset$ {no fat arc has been found}

3.1 Outline

The algorithm `FatArcSegment` (see Alg. 1) is based on the corresponding techniques in the parametric case, but it uses the bounds which are obtained with the help of Lemma 1. It assumes that the conditions of Lemma 2 are satisfied.

The algorithm is successful, if it finds three points $\mathbf{b}_1, \mathbf{b}_2, \mathbf{c}$ of the median arc, the fat arc thickness is smaller then the prescribed tolerance ε and if there is no sign change of f along the boundaries of the fat arc. It the returns the fat arc which bounds the curve segment.

It may happen, that there are no fat arc boundaries, or only one of the bounding arcs can be generated (e.g. when the distance bound of the median arc and the implicitly defined curve is greater then the radius of the meridian circle, or one of the bounding arcs does not intersect the box). The local algorithm fails if no fat arcs are generated. The algorithm then returns the empty set.

Figure 2 presents three examples of fat arcs which have been generated with the help of the algorithm. The individual steps of the algorithm are described in the next section.

3.2 Details

We describe the generation of approximate points of the curve (lines 1 and 2 of the algorithm), the segmentation of the fat arc boundaries and the sign change analysis along the segmented boundaries (line 9).

Approximate intersection points In order to construct the median arc (line 5), we approximate three points of the implicitly defined curve. Two of them are the intersections with the boundaries of the box, while the third point is the intersection with the bisector of the first two points.

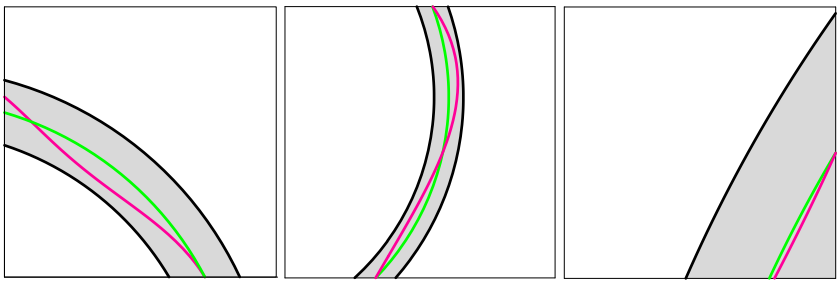

Fig. 2. Examples for fat arc generation with the help of algorithm `FatArcSegment`.The red curves are the implicitly defined curves. The median circles are shown in green.

In the case of a corner event, the corner is a boundary point of the curve. In the case of an edge event, the corresponding edge contains an intersection of the curve with the boundary of the box. It is then approximated as follows: We consider the restriction of f to the edge and generate its best L^2 approximation by a quadratic polynomial q^* which additionally interpolates the values of f at the two neighboring corners. The root of q^* then defines the approximate intersection of $\mathcal{F}$ with the edge. If no simultaneous corner event occurs at the neighbouring two corner points, then there is exactly one root, since the BB coefficients of f possess exactly one sign change from plus to minus or vice versa.

After generating the first two points, we restrict the function f to the intersection of the bisector with the box. Again we generate its best L^2 approximation by a quadratic polynomial q^* which additionally interpolates the values of f at the two end points. The root of q^* then defines the approximate intersection of $\mathcal{F}$ with the bisector.

Using similar arguments as in [18] it can be shown that the distance of the approximate points to the curve behaves as $\mathcal{O}(h^3)$, where h is the size of the box in global coordinates, cf. Eq. (3).

Analysis of fat arc boundaries The fat arc in line 10 of the algorithm is obtained by intersecting the circular ring $\mathcal{C}_d$ with the box. Its boundaries consists of segments of circular arcs and of segments of box boundaries. See Fig. 3 for several examples. We create the segments of the circular arcs and represent them by rational quadratic Bézier curves. These segments will be referred to as the boundaries of the fat arc.

Since the diameter d of the fat arc is only a bound on the one–sided Hausdorff distance $\mathrm{HD}(\mathcal{G}, \mathcal{F}^*)$, we need to verify that the fat arc indeed contains the implicitly defined curve segment. This is done by analyzing the sign of f along the fat arc boundaries.

For each segment of the fat arc boundary, we compose f with the quadratic rational parametrization and obtain a rational function in Bernstein-Bézier form.

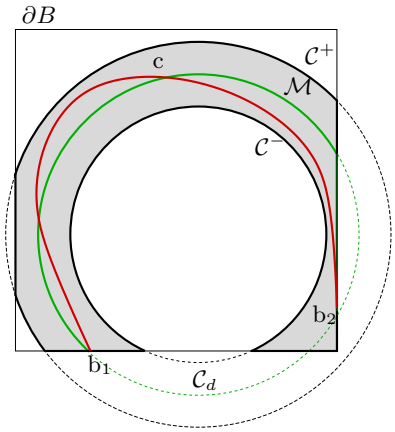

Fig. 3. Fat arc segmentation with a box: the thick black curves are the segmented fat arc boundaries

If all weights and all coefficients of the numerator have the same sign, respectively, then no sign changes of f are present (see line 9 of the algorithm).

3.3 Approximation order

Since the approximation order of curves by segments of circular arcs is three (see [2]), the same result is anticipated for the results produced by algorithm `FatArcSegment`. We confirm this approximation order by numerical examples. Consider the three bivariate polynomials

$$\begin{aligned} f_1(\mathbf{x}) &= x_1^4 + x_1^3 x_2^2 + 2x_1^2 x_2 - 6x_1 x_2 + x_2^4 - 8x_2^2 - 12x_2 \\ f_2(\mathbf{x}) &= -x_1^3 - x_1^2 x_2 + x_1 x_2 - x_2^3 + x_2^2 - 2x_2 \\ f_3(\mathbf{x}) &= -4x_1^3 - 5x_1^2 + 2x_2 \end{aligned} \tag{14}$$

with the domains (in global coordinates)

$$B^k = [-10^{-k}, 10^{-k}] \times [-10^{-k}, 10^{-k}], \quad k \in \mathbb{R}. \tag{15}$$

In the case of the first polynomial, Fig. 4 shows the result of the fat arc constructions for several values of k. The implicitly defined curve is the red one, the median arc denoted with green and the fat arcs are represented with black.

Fig. 5 visualizes the relation between the width of the fat arc and the size of the box for the three polynomials in (14). For sufficiently large values of k, the slopes of the three curves in the doubly-logarithmic plot are all three, thus confirming the expected approximation order.

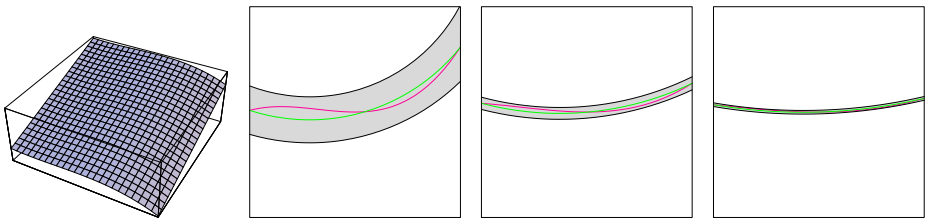

Fig. 4. Left: The graph of f_1. Right: Fat arcs for $k = 0.5, 0.75, 1.0$

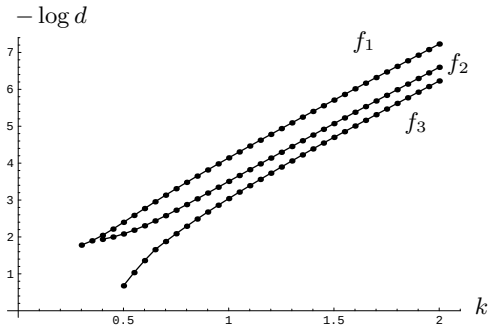

Fig. 5. Dependency between diameter and box size

4 Approximation of an implicitly defined curve

We describe an algorithm which generates a collection of fat arcs bounding a given implicitly defined curve in a box. We present several examples and compare the performance of fat arcs with bounding boxes.

4.1 The global algorithm

The algorithm `GenerateFatArcs` (see Alg. 2) combines `FatArcSegment` with recursive subdivision. First it analyzes the signs of the Bernstein–Bézier coefficients with respect to the current box. If no sign changes are present, then the current box does not contain any components of the implicitly defined curve. Otherwise it tries to apply the algorithm for a single segment. If this is not successful, then the algorithm either subdivides the current box into four squares or returns the entire box, if its diameter is already below the user-defined threshold ε.

Note that the algorithm may return boxes which do not contain any segments of the implicitly defined curve ("false positive boxes"). However, it is guaranteed that it returns a region which contains the implicitly defined curve.

4.2 Examples

We illustrate the performance of the algorithm by four examples.

Algorithm 2 `GenerateFatArcs`(f, B, ε)

if $\min d_{ij} > 0$ or $\max d_{ij} < 0$ **then**
  return $\emptyset$ {the box is empty}
end if
if f satisfies the assumptions of Lemma 2 **then**
  $\mathcal{A} \leftarrow$ `FatArcSegment`(f, B, ε) {single fat arc generation}
  if $\mathcal{A} \neq \emptyset$ **then**
    return $\mathcal{A}$ {... has been successful}
  end if
end if
if diameter of $B > \varepsilon$ **then**
  subdivide the box into 4 subboxes $B_1, \ldots, B_4$ {quadsection}
  return $\bigcup_{i=1}^{4}$ `GenerateFatArcs`(f, B_i, ε) {recursive call}
end if
return B {current box is small enough}

Example 1. The first example (see Fig. 6) visualizes the entire algorithm. We apply the algorithm to a bivariate polynomial of degree $(1, 4)$ which has only one arc in the region of interest and choose a relatively large tolerance ε. The first call of the algorithm produces four subboxes which are then analyzed independently. The first box contains an arc which can be approximated by a single fat arc. The second box produces other four subboxes, while the third and the fourth boxes do not contain any points of the implicitly defined curve. Finally, analyzing the four second-generation subboxes leads to three additional fat arcs and one empty box. The output is generated by collecting all boxes in the leaves of the tree.

Example 2. We consider a polynomial f of degree (6,9) with randomly generated BB coefficients in $[-1, 1]$. Figure 7(a) shows the surface and the implicitly defined curve segments. Figure 7(b) and (c) demonstrate the behavior of the algorithm for different tolerances ε. The upper row shows the entire domain, while the lower row shows a zoomed view of the lower left corner of the box. In the case of $\varepsilon = 0.1$, which is shown in (b), some boxes are returned as bounding boxes, since `FatArcSegment` fails and the diameter of the boxes is smaller than ε. For the smaller value of $\varepsilon = 0.01$, the fat arc generation succeeded in all generated boxes.

In the next two examples we compare fat arcs with (recursively generated) bounding boxes. In the latter case we accepted boxes with a diameter less than the prescribed tolerance.

Example 3. We approximate an implicitly defined curve, see Figure 8, by fat arcs (a) and by bounding boxes (b). Clearly, the use of fat arcs leads to a much smaller number of bounding geometric primitives. This becomes even more dramatic for smaller tolerances. The plot in (c) shows the relation between the number of generated primitives (fat arcs or boxes) and the tolerance $\varepsilon = \sqrt{2}/2^k$.

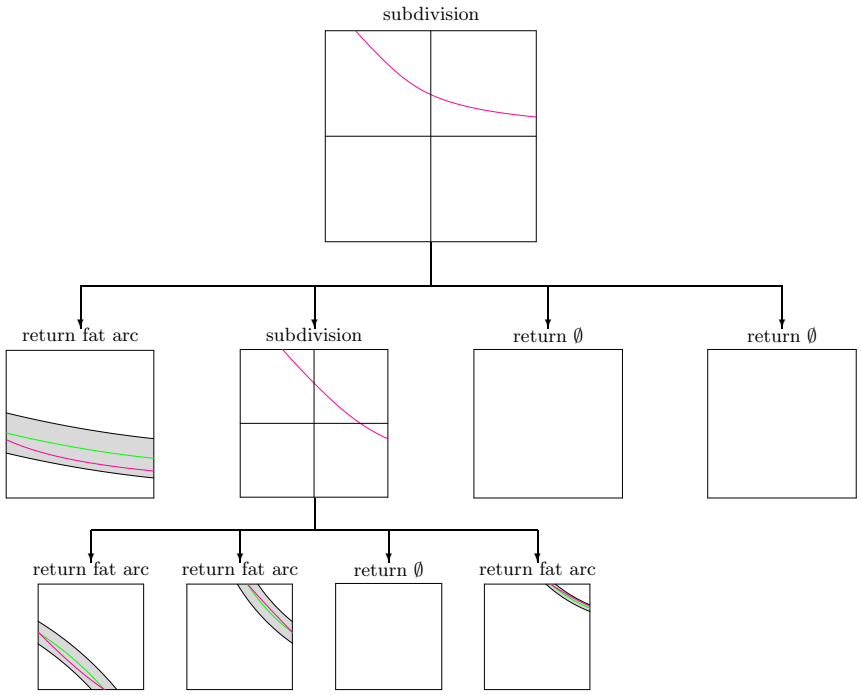

Fig. 6. Example 1: The decision tree of algorithm `GenerateFatArcs`.

Example 4. This example is based on an implicitly defined curve which possesses a singular point, see Figure 9. In this situation, the fat arc generation will fail for any box which contains the singular point, since no positive lower bound c on $||\nabla f||$ exists. Consequently, the algorithm always returns a box containing this point. Still, the results generated by our method (left) compare favorably with the use of bounding boxes (right).

5 Approximation of surface-surface intersections

As an application we apply the fat arc generation algorithm to the intersection problem between implicitly and parametrically defined surfaces. The computation of surface-surface intersections is a potential application of bounding region generation methods. In practice, intersection computation of a parametric and an implicitly defined surface is one of the most frequently encountered cases [19]. A good survey is given by [20, 21] in this topic.

Consider an implicitly defined surface $h(x, y, z) = 0$ and a parametric surface patch $\mathbf{r}(\xi_1, \xi_2)$ with domain $\Omega = [0, 1]^2$. Then the implicitly defined curve $f = h \circ r = 0$ describes the intersection curve in the domain of the parametric surface patch.

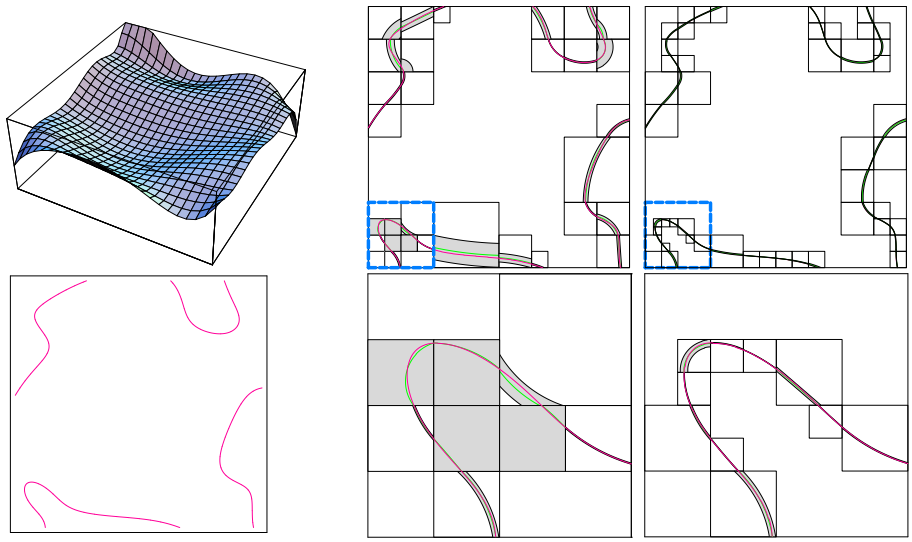

Fig. 7. Example 2: Fat arc generation for different tolerances. The graph of f and the implicitly defined curve (a), and The fat arcs (top) and a zoomed view (bottom) for $\varepsilon = 0.1$ (b) and for $\varepsilon = 0.01$ (c).

Using Algorithm `GenerateFatArcs` one may now construct a collection of fat arcs with maximum width ε in Ω. The region described by them corresponds to a certain subset (a strip) on the parametric surface patch.

Recall that the coefficients of the first fundamental form are defined as

$$g_{ij}(\xi_1, \xi_2) = \frac{\partial}{\partial \xi_i}\mathbf{r}(\xi_1, \xi_2) \cdot \frac{\partial}{\partial \xi_j}\mathbf{r}(\xi_1, \xi_2). \tag{16}$$

In order to relate the thickness of the bounding fat arcs to the thickness of the corresponding strip on the parametric surface, we present the following observation.

Lemma 3. *We assume that there exists a constant C such that*

$$g_{11}(\xi_1, \xi_2)\cos^2\phi + 2g_{12}(\xi_1, \xi_2)\cos\phi\sin\phi + g_{22}(\xi_1, \xi_2)\sin^2\phi \leq C \tag{17}$$

holds for all $\phi \in [0, 2\pi]$ and $(\xi_1, \xi_2) \in \Omega^0$. Consider a single fat arc with width ε in the parameter domain. Then the width of the corresponding region on the parametric surface patch is bounded by $2\varepsilon\sqrt{C}$.

Proof. The length L of a curve on the surface which corresponds to any straight line segment

$$(\xi_1(t), \xi_2(t)) = (\xi_1^0, \xi_2^0) + t(\cos\phi, \sin\phi), \quad t \in [a, b] \tag{18}$$

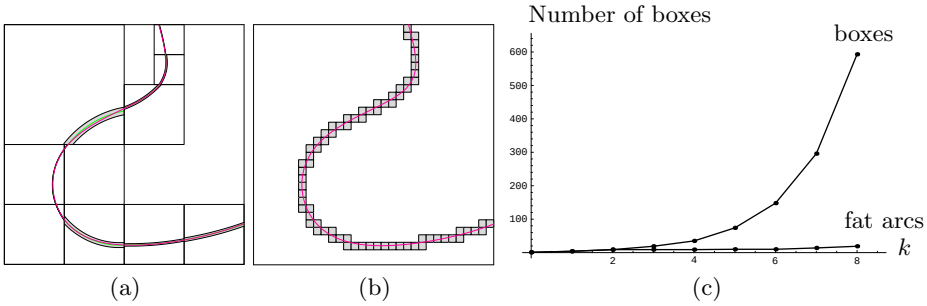

Fig. 8. Example 3: Comparison of fat arcs (a) and bounding boxes (b). The relation between tolerance and number of bounding primitives (c)

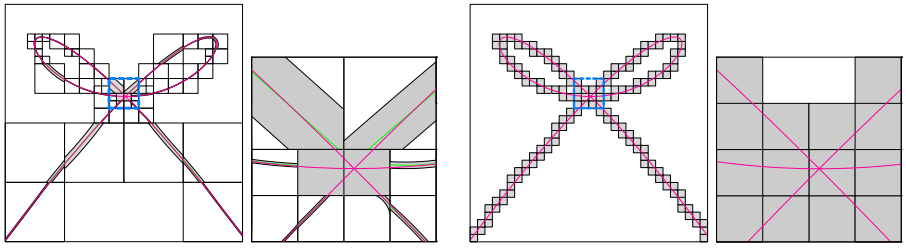

Fig. 9. Example 4: Fat arcs (left) and bounding boxed (right) for an implicitly defined curve with a singular point, where $\varepsilon = \sqrt{2}/2^5$.

in the parameter domain satisfies

$$L = \int_a^b \sqrt{g_{11}\cos^2\phi + 2g_{12}\cos\phi\sin\phi + g_{22}\sin^2\phi}\;\mathrm{d}t \le (b-a)\sqrt{C}. \qquad (19)$$

This observation can now be applied to the lines which pass through the center of the fat arc.

If $\mathbf{r}$ is a rational surface patch, then the constant C in (17) can be found by found by replacing $(\cos\phi, \sin\phi)$ by a rational parametrization of a semicircle[1], say with a parameter $\tau \in [0,1]$. Then the left–hand side of (17) defines a trivariate rational function which depends on the three parameters $(\xi_1, \xi_2, \tau) \in [0,1]^3$ and the bound C can be found with the help of the rational BB representation.

Example 5. We consider the intersection of a cubic implicitly defined surface with a biquadratic surface patch. Fig. 10, upper row, shows the intersecting surfaces and the implicitly defined intersection curve in the parameter domain.

[1] A semicircle is sufficient, due to the symmetry of the quadratic form in (17).

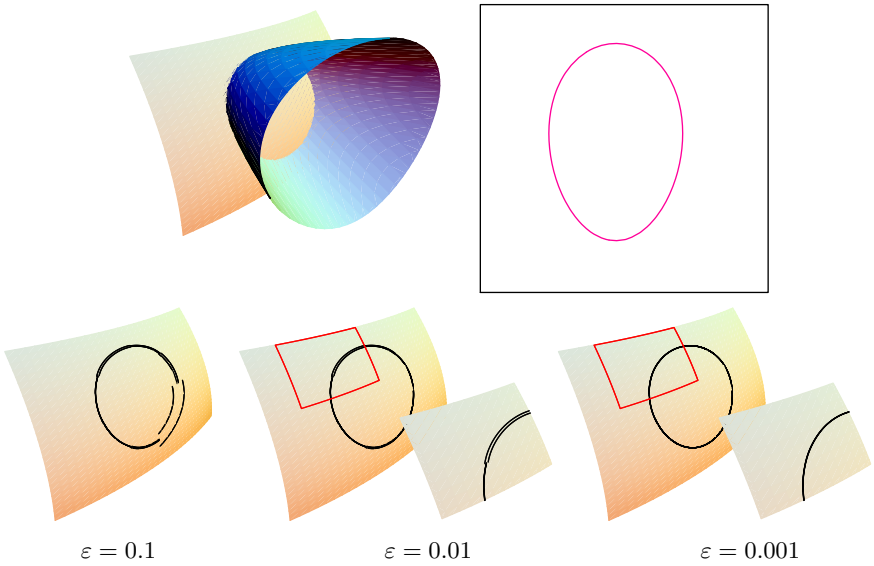

Fig. 10. Example 5: Intersection of a cubic implicit and a biquadratic parametric surface, represented by fat arcs in the parameter domain. The number of fat arcs grows from 10 for $\varepsilon = 0.1$ to 25 for $\varepsilon = 0.01$. For the larger two tolerances, we also zoomed into a segment of the surface patch.

The lower row shows the region on the surface which correspond to fat arcs in the parameter domain for three different values of the tolerance ε.

6 Conclusion

We have presented an algorithm which generate a collection of bounding fat arcs for a given planar curve. In contrast to the existing techniques, which mostly assume that parametric representations are given, the algorithm can be applied to implicitly defined curves.

The planned future work includes the extension to surfaces and to space curves, which can be bounded by toroidal surface segments. We also plan to use the results for solving systems of polynomial equations, where bounding primitives of higher approximation order may help to accelerate the convergence [18, 22]. Finally it might also be interesting to apply the algorithm to the problem of analyzing and certifying the topology of an algebraic curve, see e.g. [23].

As another possible extension, one might consider other types of median curves, such as conic sections or algebraic curves of degree higher than two. While this may provide an even higher rate of convergence (cf. [24]), it makes the generation of the fat curve more difficult (since e.g. the class of conic sections, unlike circular arcs, is not closed under offsetting). As an advantage, the use of circular arcs leads to simpler algorithms for intersection computation.

References

1. Ballard, D.H.: Strip trees: a hierarchical representation for curves. Commun. ACM **24**(5) (1981) 310–321
2. Sederberg, T.W., White, S.C., Zundel, A.K.: Fat arcs: a bounding region with cubic convergence. Comput. Aided Geom. Des. **6**(3) (1989) 205–218
3. Yang, X.: Efficient circular arc interpolation based on active tolerance control. Computer-Aided Design **34**(13) (2002) 1037–1046
4. Sabin, M.A.: The use of circular arcs for the interpolation of curves through empirical data points. Technical Report VTO/MS/164, British Aircraft Corporation (1976)
5. Marciniak, K., Putz, B.: Approximation of spirals by piecewise curves of fewest circular arc segments. Computer-Aided Design **16**(2) (1984) 87–90
6. Qiu, H., Cheng, K., Li, Y.: Optimal circular arc interpolation for NC tool path generation in curve contour manufacturing. Computer-Aided Design **29**(11) (1997) 751–760
7. Walton, D.S., Meek, D.J.: Approximating quadratic NURBS curves by arc splines. Computer-Aided Design **25**(6) (1993) 371–376
8. Walton, D.S., Meek, D.J.: An arc spline approximation to a clothoid. Journal of Computational and Applied Mathematics **170**(1) (2004) 59–77
9. Walton, D.S., Meek, D.J.: Spiral arc spline approximation to a planar spiral. Journal of Computational and Applied Mathematics **170**(1) (2004) 59–77
10. Yong, J.H., Hu, S.M., Sun, J.G.: Bisection algorithms for approximating quadratic bézier curves by G^1 arc splines. Computer-Aided Design **32**(4) (2000) 253–260
11. Šír, Z., Feichtinger, R., Jüttler, B.: Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. Comp.–Aided Design **38** (2006) 608–618
12. Held, M., Eibl, J.: Biarc approximation of polygons within asymmetric tolerance bands. Computer-Aided Design **37**(4) (2004) 357–371
13. Lin, Q., Rokne, J.G.: Approximation by fat arcs and fat biarcs. Computer-Aided Design **34**(13) (2002) 969–979
14. Aigner, M., Szilagyi, I., Schicho, J., Jüttler, B.: Implicitization and distance bounds. In Mourrain, B., Elkadi, M., Piene, R., eds.: Algebraic Geometry and Geometric Modeling. Springer (2006) 71–86
15. Sederberg, T.W.: Planar piecewise algebraic curves. Comp. Aided Geom. Design **1** (1984) 241–255
16. Hoschek, J., Lasser, D.: Fundamentals of Computer Aided Geometric Design. AK Peters, Wellesley, Mass. (1993)
17. Xu, G., Bajaj, C.L., Xue, W.: Regular algebraic curve segments (i) – definitions and characteristics. Comput. Aided Geom. Des. **17**(6) (2000) 485–501
18. Bartoň, M., Jüttler, B.: Computing roots of polynomials by quadratic clipping. Comp. Aided Geom. Design **24** (2007) 125–141
19. Lee, K.: Principles of CAD/CAM/CAE Systems. Addison-Wesely (1999)
20. Pratt, M.J., Geisow, A.D.: Surface/surface intersection problem. In Gregory, J., ed.: The Mathematics of Surfaces II. Claredon Press, Oxford (1986) 117–142
21. Patrikalakis, N.M., Maekawa, T.: Shape interrogation for computer aided design and manufacturing. Springer (2002)
22. Mourrain, B., Pavone, J.P.: Subdivision methods for solving polynomial equations. Journal of Symbolic Computation (2008) accepted manuscript.

23. Gonzalez-Vega, L., Necula, I.: Efficient topology determination of implicitly defined algebraic plane curves. Comput. Aided Geom. Design (2002) 719–743
24. Dokken, T.: Approximate implicitization. In Lyche, T., Schumaker, L., eds.: Mathematical methods for curves and surfaces. Vanderbilt University Press, Nashville (2001) 81–102

Technical Reports of the Doctoral Program "Computational Mathematics"

2009

2009-01 S. Takacs, W. Zulehner: *Multigrid Methods for Elliptic Optimal Control Problems with Neumann Boundary Control* October 2009. Eds.: U. Langer, J. Schicho

2009-02 P. Paule, S. Radu: *A Proof of Sellers' Conjecture* October 2009. Eds.: V. Pillwein, F. Winkler

2009-03 K. Kohl, F. Stan: *An Algorithmic Approach to the Mellin Transform Method* November 2009. Eds.: P. Paule, V. Pillwein

2009-04 L.X.Chau Ngo: *Rational general solutions of first order non-autonomous parametric ODEs* November 2009. Eds.: F. Winkler, P. Paule

2009-05 L.X.Chau Ngo: *A criterion for existence of rational general solutions of planar systems of ODEs* November 2009. Eds.: F. Winkler, P. Paule

2009-06 M. Bartoň, B. Jüttler, W. Wang: *Construction of Rational Curves with Rational Rotation-Minimizing Frames via MöbiusTransformations* November 2009. Eds.: J. Schicho, W. Zulehner

2009-07 M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, A.V. Vuong: *Swept Volume Parameterization for Isogeometric Analysis* November 2009. Eds.: J. Schicho, W. Zulehner

2009-08 S. Béla, B. Jüttler: *Fat arcs for implicitly defined curves* November 2009. Eds.: J. Schicho, W. Zulehner

Doctoral Program
"Computational Mathematics"

Director:

Prof. Dr. Peter Paule
Research Institute for Symbolic Computation

Deputy Director:

Prof. Dr. Bert Jüttler
Institute of Applied Geometry

Address:

Johannes Kepler University Linz
Doctoral Program "Computational Mathematics"
Altenbergerstr. 69
A-4040 Linz
Austria
Tel.: ++43 732-2468-7174

E-Mail:

office@dk-compmath.jku.at

Homepage:

http://www.dk-compmath.jku.at

Submissions to the DK-Report Series are sent to two members of the Editorial Board who communicate their decision to the Managing Editor.